International Journal of Robotics and Automation (IJRA)
Vol. 3, No. 2, June 2014, pp. 112~117
ISSN: 2089-4856

Rolling Path Plan of Mobile Robot Based on automatic diffluent ant algorithm

Zhou feng
Shandong Institute of Commerce and Technology, Shang Dong JiNan, 250103

Article Info

Article history:

Received Aug 11, 2013
Revised Mar 4, 2014
Accepted Mar 26, 2014

Keyword:

Ant algorithm
Automatic diffluent
Path planning
Rolling plan

ABSTRACT

This paper proposes a new rolling algorithm for path plan of mobile robot based on automatically shunt the ant algorithm. The goal node is mapped to the node nearby the boundary in the eyeshot of the robot, and make out the best partial path, which the robot goes ahead for a step. The algorithm will iterate one time when the robot arrives at the goal node. The robot will reach the destination along the optimal path. Simulation experiments illustrate that the algorithm can be used to plan the optimal path for mobile robot even in the complex and unknown static environment.

Corresponding Author:

Zhou feng
Shandong Institute of Commerce and Technology, Shang Dong JiNan, 250103
Email: zhoufengkey@163.com

1. INTRODUCTION

Path planning is an basic issue in researching robotics. It is also an important branch of this field. It means that In an environment with obstacles, how to find a suitable collision-free path for a mobile robot to move from a start location to a target location. Robot path planning has been an active research area, and many methods have been developed to tackle this problem, such as rolling plan [1], RRT methods [2], neural networks approaches [3], GA methods [4] and so on. Ant system (AS) algorithm is a novel simulated evolutionary algorithm. In recent years, many scholars have applied AS algorithm to the path planning of mobile robots, but the common problem is that their algorithm limited on the original AS algorithm. In the original AS algorithm, Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction rich in pheromone rather than a poor one. Due to this positive feedback, the algorithm can’t broader search, so the algorithm is easy early convergence. Latest research shows that automatic diffluence is the characteristic of real ants [7]. If the path have lots of ants which means a heavy traffic, the later ant will choose another path to reach the destination they want

In this paper, a new method for robot path planning is proposed based on automatic diffluent ant algorithm. First the grid method is built to describe the working space of the mobile robot, the goal node is mapped to the node nearby the boundary in the eyes hot of the robot, and the best local path is planned with automatic diffluent ant algorithm, when the robot goes ahead for a step. The algorithmwill iterate when the robot arrives at the goal node. According to the experiments, the method is effective.

2. DESCRIPTION OF THE ENVIRONMENT

This paper uses Grid method modeling. For convenience, we name the robot *Rob*. Let δ be the length that the *Rob* can move freely at one time.At any time, *Rob* can probe the environmental information in a circle, which radius is *r* and center is Rob's current position. This area is named *RView*.The velocity of *Rob* is $\boldsymbol{V_R}$, and the time spent on probing the environment and planning the path can be ignored. Let *AS* be a finite 2-D field, in which a finite number of static obstacles $Sb_1, Sb_2, \cdots, Sb_n$. $\sum_0$ is the right-angled coordinate system in *AS* , its origin is the left and upper corner of *AS* and its landscape orientation is X-axes, its portrait is Y-axes. The maximal values of *x* and *y* are $x_{\max}$ and $y_{\max}$, respectively. Then *x* and *y* can be divided using *δ* as an unit, as a result, all grids can be formed one by one (figure 1). The numbers of each row and column are $N_x = x_{\max} / R_a$ and $N_y = y_{\max} / R_a$, respectively.The mobile robot environment is represented by orderly numbered grids （*C*={1,2,3,…， ,*M*}）, each of which represents a location in the environment. *g*(*x* ,*y*) also represents a location in the environment, *x* is the row number,*y* is the columns number.

In the grid based robot environment, the number *i* and *g*(*x*,*y*) denote grid and environment coordinate, respectively,thus

$$x_i=((i-1) \bmod N\text{x})+1, \quad y_i=(\text{int})((i-1)/N\text{x})+1 \tag{1}$$

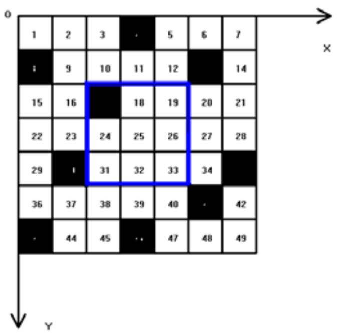

Figure 1. Relationship between grid coordinates and sequence number

3. ROLLING PATH PLANNING BASED ON AUTOMATIC DIFFLUENT ANT ALGORITHM

3.1. Thought of Algorithm and Definition

Latest research shows that automatic diffluent is the behavior characteristic of real ants [7]. Ants have the ability to automatic distribution, just like high-speeding cars. Many ants in the same path will cause traffic jam. Therefore, if there are many ants in the same path, those who come later will chose another path. The algorithm designed in this paper is based on ants' characteristic. When a node already had been chosen by many ants, those ants who came later will divert to other nodes automatically. And it can enlarge the search range, increase the diversity of the solution, in order to search everywhere to get the best solution.

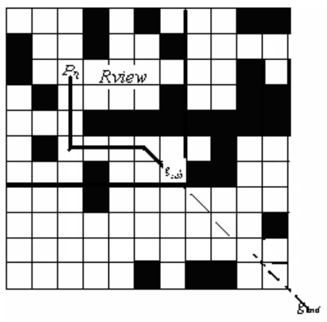

Figure 2. The sub-goal schematic diagram

Rob's current position is $P_R(t)$. The goal node is mapped to the node nearby the boundary in the eyeshot of the robot, as a local sub-targets g_{sub}, *Rob* detect the environmental information of *RView*, and the best local path form $P_R(t)$ to g_{sub} is planned by automatic diffluent ant algorithm, then the robot goes ahead for a step along this path.*Rob* goes forward every time and then carries on the above local path planning process. Therefore, the path is real-time dynamically modified. When g_{sub} and g_{end} are (in) the same position, *Rob* goes forward directly for g_{end}. In this algorithm, *Rob* always goes forward to the global object, furthermore the local path is optimal. The algorithm iterates when *Rob* arrives at the goal node. Figure 2 is a schematic diagram that shows how g_{sub} is mapped to the sub-goal.
In order to describe conveniently, We make the following definition:
Definition 1. $\mathrm{counter}(g_i,g_j)$ means the number of ants which visited the line between g_i and g_j. $1 \le \mathrm{counter}(g_i,g_j) \le CMAX$. When the $\mathrm{counter}(g_i,g_j)$ is the larger, the probability of which selected is smaller.
Definition 2. *Movemax* is the maximum of any ant's step. In order to prevents the ant searching limitlessly.
Definition 3. If $\forall g \in A$, $g \notin SS$, g is called feasible node. All feasible nodes make up the feasible region, Marked as *FS.* If $\forall g \in A$, $g \in SS$, g is called the forbidden node. All forbidden nodes make up the forbidden region, Marked as *NFS.*
Definition 4. The distance between any two grid cells g_i and g_h or corresponding points P_i and P_h is the length of the line between the center points of the two grids and (which) is denoted by $d(g_i, g_h)$or $d(P_i, P_h)$. $i,h \in C$.

$$d(g_i,\ g_h) = \sqrt{(x_i-x_h)^2+(y_i-y_h)^2} \tag{2}$$

Definition 5. If g is a grid cell, the set $BR_i(g_i(x_i,y_i))=\{g|g\in A, d(g,g_i)\le\sqrt{2}\}, i\in C$ is called the neighborhood of g_i. The broad lines marked the neighborhood of $g(4,4)$ in figure 1.
Definition 6. Let k be an ant, $tabu_k$ is the set of grid which have been passed by k from time t_0 to t_i. $tabu_k$ is called the forbidden table. $tabu_k = \{P(t_0), P(t_1), \ldots, P(t_i)\}$.

The path taken by k from time t_0 through time t_i is $path_k = \{P_1, P_2, \cdots, P_i\}$ .The sequence of connection lines between adjacent grids in $path_k$ is called the path from P_0 to P_i. The length, L, of the path is given by formula (3):

$$L=\sum_{l=1}^{e} d_l \quad d_l=d(g_i, g_h),\ g_i, g_h \notin Os, i,h\in C \tag{3}$$

Definition 7. The heuristic function which guides the ant to choose the next node during its search is called $\eta(g)$, that g is an arbitrary grid cell. In this paper $\eta(g) = 1/d(g, g_{sub})$ for ant's family 1 and $\eta(g) = 1/d(g, g_R)$ for ant's family 2.
Definition 8. Suppose $k_1 \in \mathrm{ant}_1$ and $k_2 \in \mathrm{ant}_2$. k_1 starts from g_{begin} and k_2 from g_{end}. At time t_n the position of k_1 is P_{k_1}, the position of k_2 is P_{k_2} If $|d(P_{k_1}, P_{k_2})|=0$,we say that k_1 and k_2 have encountered each other.
Definition 9. $RView(P_R(t_i))=\{P|P\in A,\ d(P,P_R(t_i))\le r\}$ is called the visual domain of *Rob* at position $P_R(t_i)$.

3.2. Steps of the Algorithm

m ants from one family take $P_R(t)$ as nest and g_{sub} as food source, and chose the same number of ants form another family takes g_{sub} as the nest and $P_R(t)$ as the food source. These ants of two families cooperate to search for a path in an opposite direction. Each ant searches for a path by sequentially choosing the nearest node to the target point in its current neighboring domain. A random search strategy is used to increase diversity of the search. Since the algorithms for the two ant families are identical except for their different starting and ending points, the searching algorithm of ant family 1 will be taken as the example in the following and all the subscripts denoting this family will be omitted to simplify the notation.

The procedure is described as follows.

Step1: Set the starting point g_{begin} and the terminal point g_{end}, then initializes relevant parameters.

Step2: According to the method shown in Figure 2, the g_{end} point is mapped to the closest node inside *Rob's RView*; this will be regarded as the local navigation sub-goal g_{sub}. The selection of g_{sub} is governed by the following:

If $d(P_R(t_i),g_{end})=r-\delta$,then $g_{sub}=g_{end}$;otherwise

$$d(P_R(t_i),g_{sub})=r-\delta \text{ and } \min d(g_{sub},g_{end}) \qquad (4)$$

where r is the radius of *Rob*'s *RView*; and δ is *Rob*'s step length.

Step3: *Rob* detects information about obstacles with its sensor (传感器) within its current *RView*. If obstacles are detected, their coordinates（坐标）will be computed.（推算）

Step4: Assign m ants to the starting point $P_R(t)$, then add $P_R(t)$ to the forbidden table $tabu_k$ (k=1,2…m).Initializes $counter(g_i,g_j)$=1,the food-searching iteration counter n=0, *MAX*(the maximum number of iteration) and *CMAX* (the upper bound of $counter(g_i,g_j)$).

Step 5: Denote（表示）the current position of ant k is g_i ($g_i \in FS$).The ant k select the next node g_j ($g_j \in Wk_i(g_i)$, $g_j \notin tabu_k$) by Eq. (5) or Eq. (6).

$$j=\begin{cases}\arg\ \max\{[1/counter(g_i,g_j)]^{\alpha}\eta_j(g_j)^{\beta}\} & if\ \ q\le q_0\\ S & else\end{cases} \qquad (5)$$

where $j(j\in C)$is grid serial numbers of g_j. q is a random number uniformly distributed in [0,1].q_0 is a parameter（参数）(0<q_0≤1). S is a random variable selected according to the probability distribution given in Eq. (6).

$$p_{ij}^{k}(n)=\frac{[1/counter(g_i,g_j)]^{\alpha}\eta_j(g_j)^{\beta}}{\sum\limits_{q\in|Z|}[1/counter(g_i,g_j)]^{\alpha}\eta_j(g_j)^{\beta}} \quad j\notin tabu_k \qquad (6)$$

where $\eta_j(g_j)$ is computed by definition 7.□ α ($\alpha \ge 0$) expreses relative importance of $1/counter(g_i,g_j)$. β ($\beta \ge 0$) expresses relative importance of $\eta_j(g_j)$. $p_{ij}^k(n)$ is transform probability of the ant k from i to j at the n iteration. |Z| is the set of grid that remain to be visited by ant k positioned on g_j .q and q_0 prevent the stagnation. If $q>q_0$,the ant compute the probability of |Z| and select the next node g_j by roulette selection. When the number of elements is more than *Movemax* in $tabu_k$, the ant k is dead. Return to Step 4, then initializes a new ant. Or return to Step 6.

Step 6: While building a solution at g_i, ants change their pheromone level by applying the local updating rule of Eq. (7).

$$counter(g_i,g_j)=\begin{cases}counter(g_i,g_j)+1 & if \quad counter(g_i,g_j)<CMAX\\ CMAX & else\end{cases} \qquad (7)$$

Step 7: After ant k has chosen the node j, and then checks whether two ants from opposite families have met according to the conditions in definition 8. If ant has encountered another ant, then return to Step 8.Or let j take the place of i, then return to Step 5 and choose the next node. Until there is an ant which has encountered another ant, or all nodes have been chosen. a (can it be stopped.)

Step 8: If (one)ant has encountered another ant, for each new encounter connect all the cells passed by the encountering ants and compute the length L of the connected path by Eq. (3). change their pheromone on the feasible path by Eq.(8).

where Rou_1 is the pheromone decay parameter.

$$counter(g_i,g_j)=\lfloor Rou_1 * counter(g_i,g_j)\rfloor+1 \qquad (8)$$

Compute all the feasible path by Eq. (3). L_{kmin} is saved as the shortest distance. Compared L_{kmin} with the historical optimal distance L_d. If $L_{kmin} < L_d$, replace L_d with L_{kmin} and record the new set of nodes as the tentative optimal path.

Step 9: After all ants have completed their tours, the pheromone of the optimal path is updated by Eq. (9). Where Rou_2 is the pheromone global decay parameter.

$$counter(g_i, g_j) = \lfloor Rou_2 * counter(g_i, g_j) \rfloor + 1 \qquad (9)$$

Step10: Let $n=n+1$. If n<MAX, empty $tabu_k$ and return to Step 4. If $n = MAX$, the planning is finished. The last recorded path is the planned optimal path.

Step 11: If g_{end} is detected or g_{sub} is the same as g_{end}, *Rob* will walk to g_{end} along the planned path. Otherwise, *Rob* will return to step 2 after moving one step along the planned local optimal navigation path.

4. EXPERIMENT RESULTS

To investigate the effect of the algorithm proposed in this paper, many simulation experiments were conducted. This section presents the simulation results of the proposed algorithm. The proposed solution algorithm is implemented with MS Visual C++ 6.0 on a Intel Pentium4 3.0GHz computer with 256MB RAM. In these figures, shades of black are obstacles. We define that 10*10 grid as the *Rview*.

Here, we generate a background environment random (Figure 3 Figure 4). These experiments show unexpectedly excellent performance with consumed time smaller than 0.01 second, the minimum precision of the computer clock on our system.

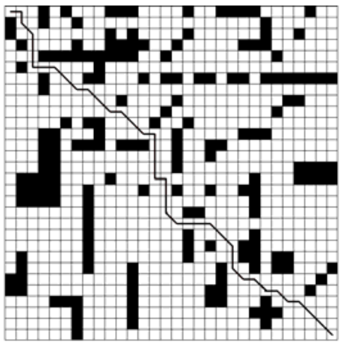
Figure 3. Optimal path is from our algorithm in simple environment

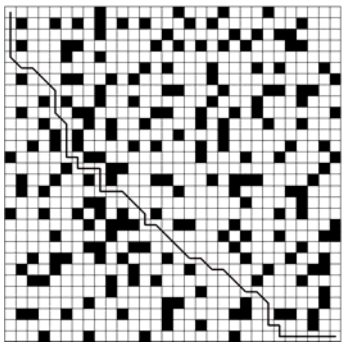
Figure 4. Optimal path is from our algorithm incomplex environment

5. SUMMARY AND CONCLUSIONS

The difficulty of navigation or path planning in complex environments with mobile obstacles and unknown static obstacles is how to guarantee the global optimal or near-optimal path and how to avoid the so called deadlock and oscillation, etc. In this paper, the final target point is mapped to a sub-goal point outside of the robot's visual domain. This sub-goal is used as the local navigation sub-goal. This process will be repeated after each of Rob's steps. Therefore every local path only constitutes a kind of navigation trend. If every local path is optimal, the global path walked by *Rob* is almost the global optimal. The direction from *Rob* to g_{end} should always be the guide for *Rob*'s direction of motion. This process is dynamically ongoing with the motion of the robot. Therefore, the robot can walk along a global optimal or almost optimal path under the guidance of the local optimal collision-free path within each visual domain of the robot and eventually reach the target point safely without collision.

REFERENCES

[1] Zhang CG, Xi YG. Path planning of robot based on rolling window in globally unknown environment. *Science in China E*. 2001; 31(1): 51-58.

[2] Dave Ferguson and Anthony Stentz. *Anytime RRTs*. Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2006: 5369-5375

[3] Xuena Qiu, Jiatao Song and Xuejun Zhang. *A Complete Coverage Path Planning Method for Mobile Robot in Uncertain Environments*. Proceedings of the 6th World Congress on Intelligent Control and Automation. 2006: 8892-8896

[4] Yanrong Hu and Simon X Yang. *A Knowledge Based Genetic A1gorithm for Path Planning of a Mobile Robot*. Proceedings of the 2004 IEEE international Conference on Robotics & Automation New Orleans, LA. 2004:4350-4355.

[5] Zhu Qing-Bao. Ant Algorithm for Navigation of Multi-Robot Movement in Unknown Environment. *Journal of Software*. 2006; 17(9): 1890-1898 (in Chinese).

[6] Ying-Tung Hsiao, Cheng-Long Chuang, Cheng-Chih Chien. Ant Colony Optimization for Best Path Planning. Int Symp on Communications and Information Technologies 2004. Japan. 2004: 109-113.

[7] Dussutour A, Fourcassie V, Helbing D, etl. Optimal traffic organization in ants under crowded conditions. *Nature*. 428(6978):70-73.